

**METHOD FOR CHANGING STATE OF REAL TIME OS**

**Publication number:** JP2001147830 (A)

**Publication date:** 2001-05-29

**Inventor(s):** YAMATO KOJI

**Applicant(s):** NEC MICROCOMP TECHNOLOGY LTD

**Classification:**

- international: G06F11/28; G06F9/46; G06F11/28; G06F9/46; (IPC1-7): G06F11/28; G06F9/46

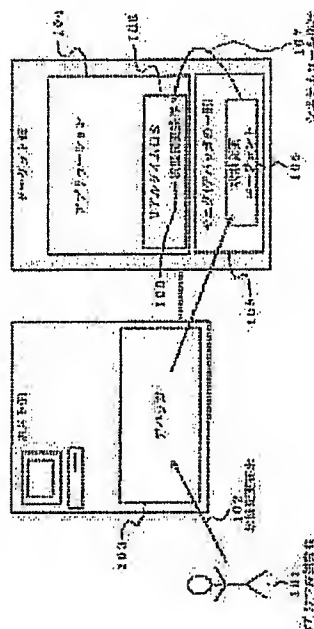
- European:

**Application number:** JP19990329320 19991119

**Priority number(s):** JP19990329320 19991119

**Abstract of JP 2001147830 (A)**

**PROBLEM TO BE SOLVED:** To provide a method for realizing the state change of a real time OS at the time of the debug of an application using the real time OS. **SOLUTION:** At the time of the debug of an application 104 operating on a real time OS 105 of a target system, when a state change request 102 of the real time OS 105 is issued, a debugger 103 allows a state change routine 106 placed on a monitor being one part of the debugger 103 on the target system to generate registered interruption. When the interruption is generated, the real time OS 105 executes the interruption pre-processing, and the state change routine 106 sets a system call parameter for changing the present value to a target value, and issues a system call, and the real time OS 105 processes the system call, and executes the state change processing.



Data supplied from the esp@cenet database — Worldwide

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-147830

(P2001-147830A)

(43)公開日 平成13年5月29日(2001.5.29)

(51) Int.Cl.<sup>7</sup>

識別記号

FI

デーポート<sup>①</sup>(参考)

G O 6 F 11/28

C O 6 F 11/28

Λ 5B042

9/46

340

9/46

340B 5B098

審査請求 有 請求項の数7 OL (全 5 頁)

(21)出願番号 特願平11-329320

(22) 出願日 平成11年11月19日(1999. 11. 19)

(71)出願人 000232151

日本電気マイコンテクノロジー株式会社  
神奈川県川崎市幸区塚越3丁目484番

(72) 發明者 大和 昇治

神奈川県川崎市幸区塚越3丁目484番 日本電気マイコンテクノロジー株式会社内

(74) 代理人 100086645

弁理士 岩佐 義幸

Fターム(参考) 5B042 GA23 HH01 NN40

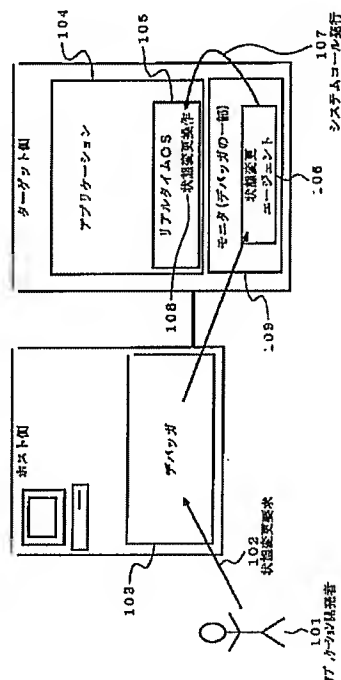
5B098 BA05 GA02 GB02 GB11 JJ06

(54) 【発明の名称】 リアルタイムOSの状態変更方法

(57) 【要約】

【課題】 リアルタイムOSを用いたアプリケーションのデバッグにおいて、リアルタイムOSの状態変更を実現する方法を提供する。

【解決手段】 ターゲットシステムのリアルタイムOS 105上で動作するアプリケーション104のデバッグにおいて、リアルタイムOS 105の状態変更要求102があると、デバッグ103は、ターゲットシステム上にありデバッグ103の一部であるモニタ上に置かれた状態変更ルーチン106が登録された割り込みを発生させ、割り込みが発生すると、リアルタイムOS 105が割り込み前処理を行い、状態変更ルーチン106が、目的の値に変更するためのシステムコールパラメータを設定すると共にシステムコールを発行し、リアルタイムOS 105がシステムコールを処理して状態の変更を行う。



【特許請求の範囲】

【請求項1】ターゲットシステムのリアルタイムOS上で動作するアプリケーションをデバッグする場合に、割り込み処理としてデバッグ上にリアルタイムOSの状態を変更する状態変更ルーチンを置き、状態変更ルーチンが、リアルタイムOSのシステムコールを利用してリアルタイムOSの状態を変更することを特徴とするリアルタイムOSの状態変更方法。

【請求項2】前記状態変更ルーチンは、ターゲットシステム上にありデバッグの一部であるモニタ上に置かれることを特徴とする請求項1に記載のリアルタイムOSの状態変更方法。

【請求項3】前記状態変更ルーチンは、ホストコンピュータのデバッグ上に置かれることを特徴とする請求項1に記載のリアルタイムOSの状態変更方法。。

【請求項4】ターゲットシステムのリアルタイムOS上で動作するアプリケーションのデバッグにおいて、リアルタイムOSの状態変更要求があると、デバッグは、ターゲットシステム上にありデバッグの一部であるモニタ上に置かれた状態変更ルーチンが登録された割り込みを発生させ、割り込みが発生すると、リアルタイムOSが割り込み前処理を行い、状態変更ルーチンが、目的の値に変更するためのシステムコールパラメータを設定すると共にシステムコールを発行し、リアルタイムOSがシステムコールを処理して状態の変更を行うことを特徴とするリアルタイムOSの状態変更方法。

【請求項5】ターゲットシステムのリアルタイムOS上で動作するアプリケーションのデバッグにおいて、リアルタイムOSの状態変更要求があると、ホストコンピュータ側のデバッグ上に置かれた状態変更ルーチンが、システムを一時的に停止し、割り込みの前処理をシミュレートし、目的の値に変更するためのシステムコールパラメータを設定し、プログラムカウンタをシステムコールの先頭アドレスに変更し、システムコールの戻りアドレスを割り込み終了部の先頭アドレスに設定し、システムを実行し、リアルタイムOSがシステムコールを処理して状態の変更を行うことを特徴とするリアルタイムOSの状態変更方法。

【請求項6】ターゲットシステムのリアルタイムOS上で動作するアプリケーションのデバッグにおいて、リアルタイムOSの状態変更要求があると、デバッグが、ターゲットシステム上にありデバッグの一部であるモニタ上に置かれた状態変更ルーチンが登録された割り込みを

発生させる手順と、

割り込みが発生すると、リアルタイムOSが割り込み前処理を行う手順と、

状態変更ルーチンが、目的の値に変更するためのシステムコールパラメータを設定すると共にシステムコールを発行する手順と、

リアルタイムOSがシステムコールを処理して状態の変更を行う手順とを実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項7】ターゲットシステムのリアルタイムOS上で動作するアプリケーションのデバッグにおいて、リアルタイムOSの状態変更要求があると、ホストコンピュータ側のデバッグ上に置かれた状態変更ルーチンが、システムを一時的に停止する手順と、

割り込みの前処理をシミュレートする手順と、

目的の値に変更するためのシステムコールパラメータを設定する手順と、

プログラムカウンタをシステムコールの先頭アドレスに変更する手順と、

システムコールの戻りアドレスを割り込み終了部の先頭アドレスに設定する手順と、

システムを実行する手順と、

リアルタイムOSがシステムコールを処理して状態の変更を行う手順とを実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、リアルタイムOSの状態の変更を実現できるリアルタイムOSの状態変更方法に関する。

【0002】

【従来の技術】従来、デバッグには、アプリケーションの変数値を変更する機能が備わっている。変数の値が期待する値と異なる場合には、変数値変更機能を用いて変数の値を変更することにより、アプリケーションの動きをデバッグすることができる。

【0003】

【発明が解決しようとする課題】しかし、この従来技術には、次のような問題点があった。その問題点とは、リアルタイムOS上のアプリケーションで、リアルタイムOSの状態変更ができないということである。

【0004】リアルタイムOS上のアプリケーションでは、処理をタスクという単位に切り分け、タスクという処理単位を、セマフォやイベントフラグといった同期メカニズムを用いて実行制御している。リアルタイムOSの状態変更とは、タスクの状態（実行、実行可能、待ち等）、セマフォ、イベントフラグの値といったものの変更である。これらは密接に関係し合っており、1つの値の変更が他の値に影響を与える。通常のデバッグにはリアルタイムOSに関する知識が存在しないため、このよ

うな関連する値を把握することはできず、リアルタイムOSの状態変更はできない。

【0005】この発明の目的は、リアルタイムOSを用いたアプリケーションのデバッグにおいて、リアルタイムOSの状態の変更を実現できるリアルタイムOSの状態変更方法を提供することにある。

【0006】

【課題を解決するための手段】この発明のリアルタイムOSの状態変更方法は、ターゲットシステムのリアルタイムOS上で動作するアプリケーションをデバッグする場合に、割り込み処理としてデバッグ内にリアルタイムOSの状態を変更する状態変更ルーチンを置き、状態変更ルーチンが、リアルタイムOSのシステムコールを利用してリアルタイムOSの状態を変更することを特徴とする。

【0007】

【発明の実施の形態】次に、この発明の実施の形態について図面を参照して説明する。

【0008】図1は、この発明のリアルタイムOSの状態変更方法についての第1の実施の形態を説明する構成図である。図1を参照すると、この発明のリアルタイムOSの状態変更方法の実施の形態は、リアルタイムOS上で動作するアプリケーションのデバッグを行うアプリケーション開発者101と、ホストコンピュータ上で動作するデバッグ103と、ターゲットシステム上にあり、デバッグの一部であるモニタ109と、モニタ10上に置かれた、リアルタイムOSの状態を変更する状態変更ルーチン106と、ターゲットシステム上にあるリアルタイムOS105と、リアルタイムOS105上で動作するアプリケーション104から構成される。

【0009】ここで、状態変更ルーチンとは、リアルタイムOSの状態を変更処理するプログラムルーチンであり、状態変更ルーチンは、ターゲットシステムの割り込み処理として登録する。

【0010】次に、図1および図2を参照して第1の実施の形態の動作について詳細に説明する。図2は、第1の実施の形態の動作を説明するフローチャートである。

【0011】まず、アプリケーション開発者101が、デバッグ103に状態変更要求102を発行する。次に、デバッグ103は、状態変更ルーチン103が登録された割り込みを発生させる（ステップ204）。割り込みが発生すると、リアルタイムOS105によって、実行タスクのコンテキストへのセーブ等の割り込み前処理が行われ（ステップ205）、その後、処理が状態変更ルーチン106に移る。そして、状態変更ルーチン106では、目的の値に変更するためのシステムコールパラメータを設定し（ステップ206）、システムコールを発行する（ステップ207）。

【0012】その後、リアルタイムOS105によりシステムコールが処理され（ステップ208）、状態の変

更108が行われる。最後に、システムコール処理終了後、リアルタイムOSの割り込み終了処理部に処理を移し（ステップ209）、ディスパッチ処理が必要な場合はここでディスパッチが行われる。以上の動作によりリアルタイムOS105上のアプリケーション104の状態変更を実行する。

【0013】次に、この発明の第2の実施の形態について説明する。図3は、この発明のリアルタイムOSの状態変更方法についての第2の実施の形態を説明する構成図である。

【0014】図1に示す実施の形態では、状態変更ルーチン106をターゲット上に置いたため、割り込み処理として動作させることができた。しかしながら、組み込み分野においては、ターゲットに十分な空きメモリがある場合というのは少ない。そこで状態変更ルーチンをターゲット上に置かず、ホスト側に置いた場合の実施の形態を説明する。

【0015】図3を参照すると、この発明のリアルタイムOSの状態変更方法の実施の形態は、リアルタイムOS305上で動作するアプリケーション304のデバッグを行うアプリケーション開発者301と、ホストコンピュータ上で動作するデバッグ303と、デバッグ303上に置かれた状態変更ルーチン306と、ターゲットシステム上にあるリアルタイムOS305と、リアルタイムOS305上で動作するアプリケーション304から構成される。

【0016】次に、図3および図4を参照して第2の実施の形態の動作について詳細に説明する。図4は、第2の実施の形態の動作を説明するフローチャートである。

【0017】まず、アプリケーション開発者301がデバッグ303に状態変更要求302を発行する。デバッグ303内の状態変更ルーチン306は、システムを一時的に停止し（ステップ404）、実行中のタスクに対するコンテキストのセーブ等の割り込みの前処理をシミュレートする（ステップ405）。次に、目的の値に変更するためのシステムコールパラメータの設定、プログラムカウンタ（PC）をシステムコールの先頭アドレスに変更し（ステップ406）、およびシステムコールの戻りアドレスを割り込み終了部の先頭アドレスに設定する（ステップ407）。

【0018】ここでシステムを実行させる（ステップ408）ことにより、リアルタイムOSによりシステムコールが処理され（ステップ409）、状態の変更が行われる。システムコールの処理終了後、リアルタイムOS305の割り込み終了処理部に処理が移り（ステップ410）、ディスパッチ処理が必要な場合はここでディスパッチが行われ、状態変更処理が終了する。

【0019】また、図3において、モニタ309は状態変更処理に関係しない。これにより、ターゲット上にモニタ309が存在しない場合でも状態変更を行うことが

可能となる。これは、ターゲットに十分な空きメモリが得られない組み込み分野のアプリケーション開発、例えば、ICEを使った場合においても、リアルタイムOSの状態変更機能を実現できることを表す。

【0020】

【発明の効果】以上説明したように、この発明は、リアルタイムOS上のアプリケーションで、リアルタイムOSの状態を変更することができる。

【0021】また、この発明は、リアルタイムOSの状態の変更によって、不具合修正、再構築、再実行といった作業をすることなく、デバッグを続けることができるため、リアルタイムOS上で動作するアプリケーションのデバッグに費やす時間を大幅に短縮することができる。

【図面の簡単な説明】

【図1】この発明のリアルタイムOSの状態変更方法についての第1の実施の形態を説明する構成図である。

【図2】第1の実施の形態の動作を説明するフローチャートである。

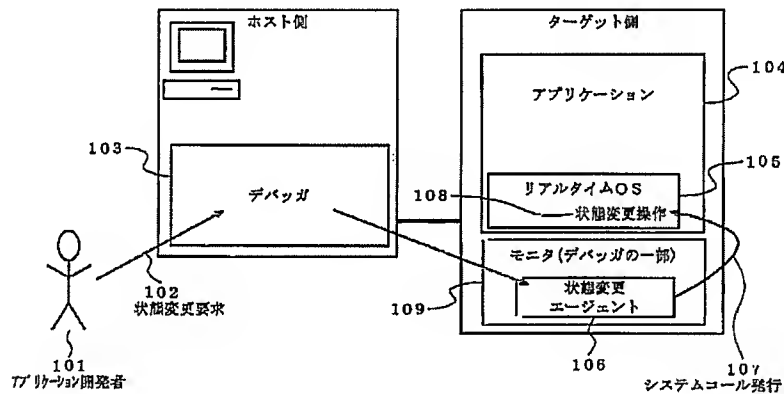
【図3】この発明のリアルタイムOSの状態変更方法についての第2の実施の形態を説明する構成図である。

【図4】第2の実施の形態の動作を説明するフローチャートである。

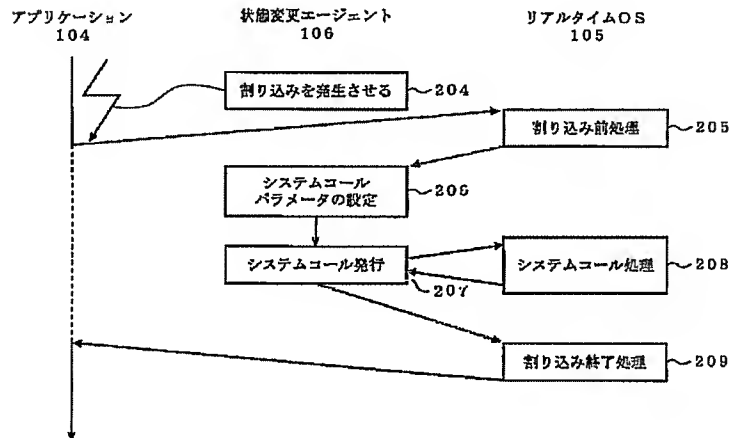
【符号の説明】

- 101, 301 アプリケーション開発者
- 102, 302 状態変更要求
- 103, 303 デバッガ
- 104, 304 アプリケーション
- 105, 305 リアルタイムOS
- 106, 306 状態変更ルーチン
- 107, 307 システムコール発行
- 108, 308 状態変更操作
- 109, 309 モニタ

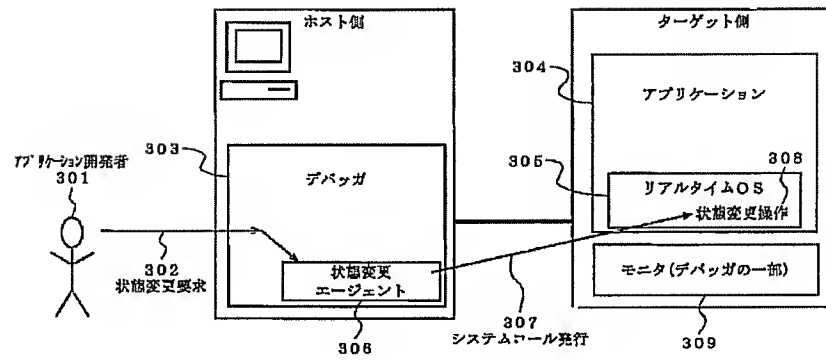
【図1】



【図2】



【図3】



【図4】

